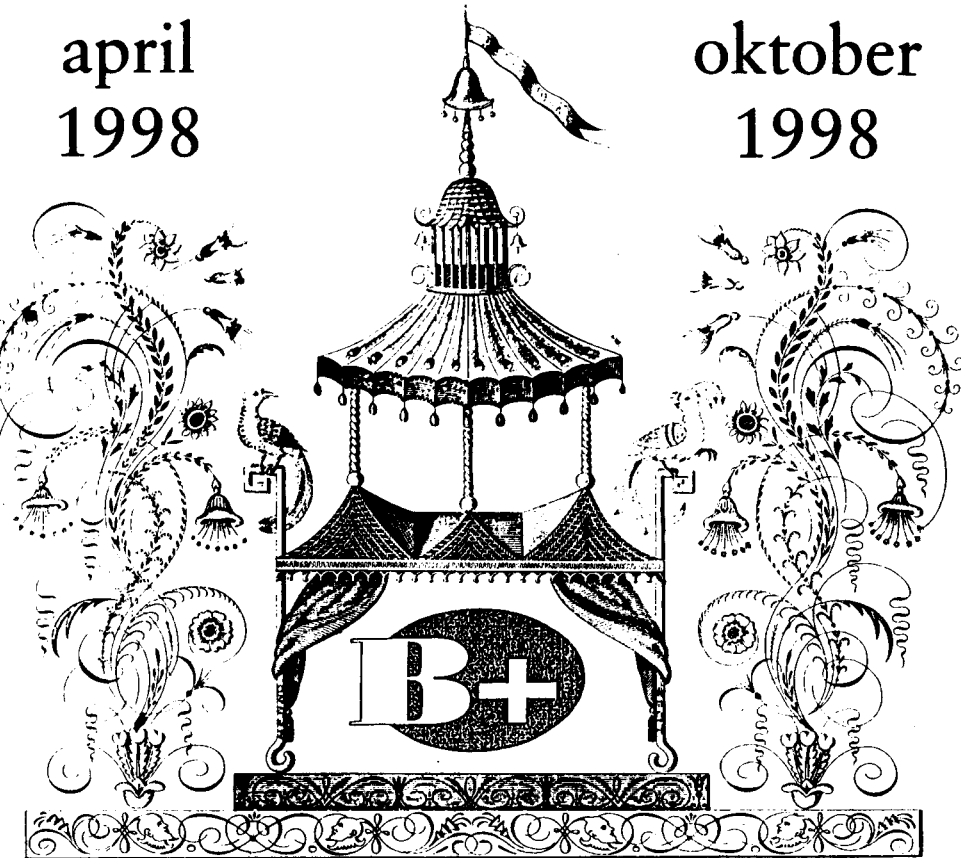
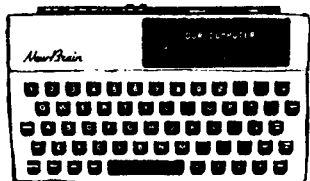


18  
april  
1998

17  
oktober  
1998



*NewBrain*dag  
nonnenwater 8, gouda



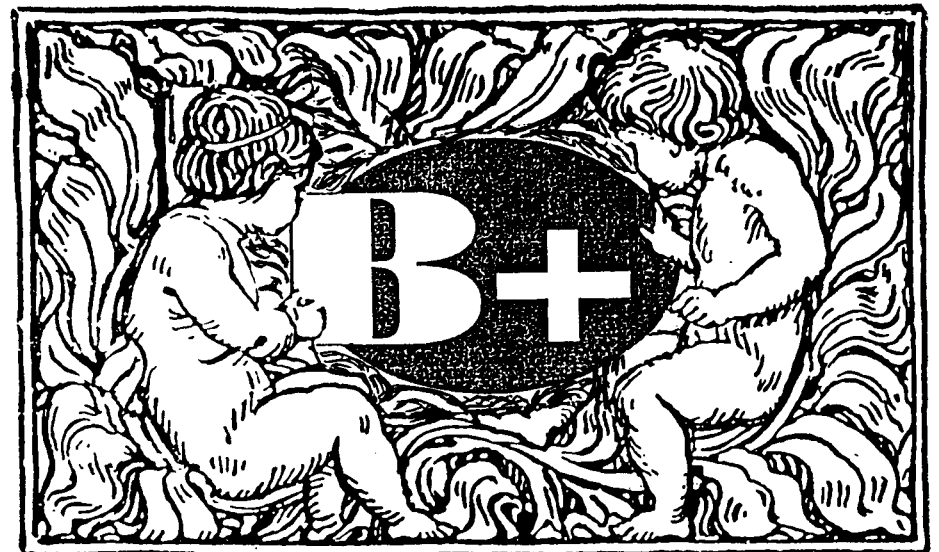
*NewBrain*-  
gebruikersgroep  
postbus 94494  
1090 GL amsterdam

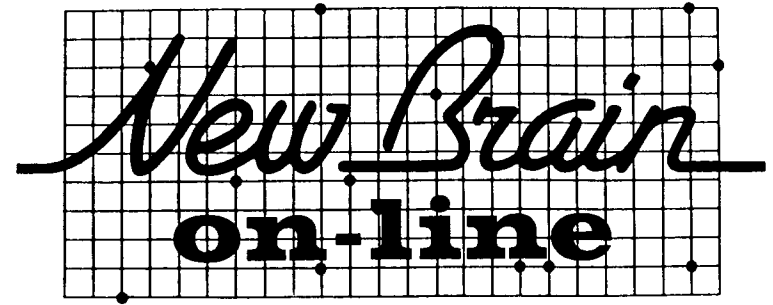
*NewBrain*  
on-line

uitgave van de  
*NewBrain* -  
gebruikersgroep

22

november 1997





ten geleide

het gaat goed met de gebruikersgroep. dat ziet u aan deze nieuwe new-brain on-line die voor u ligt. meer activiteiten in de club, meer mensen actief, een hele reeks artikelen voor uw lijfblad (en er is al kopij voor het volgende nummer), de derde serie B+-bordjes is van stapel gelopen

in dit nummer een aflevering van ton goossens' feuilleton over de micro-controller, een aantal artikelen van abraham vreugdenhil over gebruik en mogelijkheden van een lcd-scherm aan uw b+-bord, en van dré jansen achtergrondinformatie over de h-brug en de unipolaire stappenmotor

houd deze waardevolle documentatie niet voor uzelf, maar laat het ook anderen lezen. dit geldt zeker voor de artikelen van abraham vreugdenhil: hij vindt zelfs, dat de lezer verplicht is ze te kopiëren

de newbraindagen worden voortaan in gouda gehouden, in het clubhuis van de afdeling gouda. volgend jaar zijn ze op de zaterdagen 18 april en 17 oktober. op bladzijde 4 staat een routebeschrijving

menno stevens



**New-Brain-**  
gebruikersgroep  
postbus 94494  
1090 GL amsterdam



voorzitter: jan wubben, (010) 4557698  
secretaris: maarten floor, (020) 4964374  
penningmeester: menno stevens, (020) 6924137  
dré jansen, (0174) 414199  
albert stuurman, (030) 2280163

postrekening 2505800 tnv hcc newbrain-gebruikersgroep

de newbrain-gebruikersgroep is een onderdeel van de  
hcc hobby computer club  
de molen 24, 3994 DB houten  
inschrijvingsnummer kvk utrecht 82311

landelijke newbraindagen  
zaterdag 18 april 1998 • zaterdag 17 oktober 1998  
in het clubhuis van de afdeling gouda  
nonnenwater 8, 2801 VA gouda

newbrain on-line  
redactie: menno stevens, (020) 6924137  
kopij voor nummer 23 graag naar het adres van de gebruikersgroep  
of per e-mail aan mennostevens@hetnet.nl  
geplaatste artikelen mogen alleen voor niet-commerciële doeleinden,  
onder bronvermelding, worden overgenomen

# newbraindag in gouda

nu de school in leiden, waar wij de afgelopen jaren onze newbraindagen  
gehouden hebben, zich commercieel op moet stellen bij de verhuur van  
ruimten, is het tijd om te verhuizen. wij zijn blij onderdak gevonden te  
hebben bij de afdeling gouda in hun voormalige waterleidinggebouw. op  
de volgende pagina staat een routebeschrijving.  
noteer dus vast de newbraindagen van volgend jaar:

zaterdag 18 april 1998 • zaterdag 17 oktober 1998  
clubhuis van de afdeling gouda  
nonnenwater 8  
2801 VA gouda

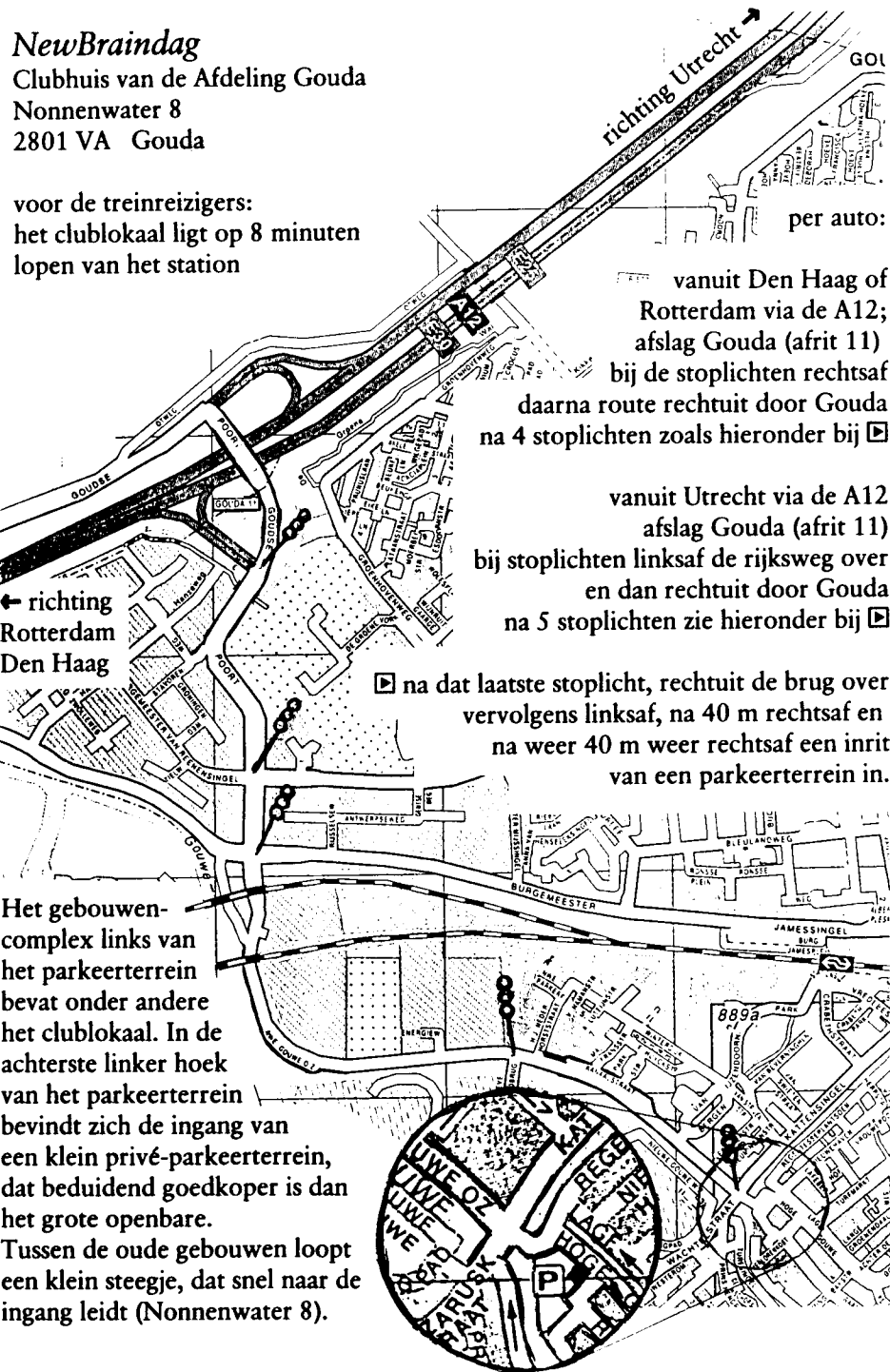
de gebruikersgroep is de directie van de sbbo erkentelijk voor het beschik-  
baar stellen van ruimte in de school gedurende de afgelopen negen jaar,  
en maarten floor voor zijn bemiddeling daarbij

menno stevens



**NewBraindag**  
Clubhuis van de Afdeling Gouda  
Nonnenwater 8  
2801 VA Gouda

voor de treinreizigers:  
het clublokaal ligt op 8 minuten  
lopen van het station



per auto:  
vanuit Den Haag of  
Rotterdam via de A12;  
afslag Gouda (afrit 11)  
bij de stoplichten rechtsaf  
daarna route rechtuit door Gouda  
na 4 stoplichten zoals hieronder bij □

vanuit Utrecht via de A12  
afslag Gouda (afrit 11)  
bij stoplichten linksaf de rijksweg over  
en dan rechtuit door Gouda  
na 5 stoplichten zie hieronder bij □

□ na dat laatste stoplicht, rechtuit de brug over  
vervolgens linksaf, na 40 m rechtsaf en  
na weer 40 m weer rechtsaf een inrit  
van een parkeerterrein in.

Het gebouwen-  
complex links van  
het parkeerterrein  
bevat onder andere  
het clublokaal. In de  
achterste linker hoek  
van het parkeerterrein  
bevindt zich de ingang van  
een klein privé-parkeerterrein,  
dat beduidend goedkoper is dan  
het grote openbare.  
Tussen de oude gebouwen loopt  
een klein steegje, dat snel naar de  
ingang leidt (Nonnenwater 8).

# de micro- controller

een serieuze poging om de DOS-gebruikers  
enig inzicht in de werking van dit soort hardware  
te verschaffen  
deel 4

de vorige delen van ton goossens' artikelenreeks over de microcontroller zijn afgedrukt in newbrain on-line 20 en 21. de serie is eerder verschenen in de softwarebus van de hcc dos-gg

Een aantal mensen schijnt wat moeite te hebben met de keuze van de processor voor B+. Inderdaad, de 80535 is een verschijnsel dat al een paar jaar oud is. Toch worden er nog veel nieuwe ontwikkelingen mee gemaakt. Moderne ontwikkelingen maken we met RISC-achtige controllers met Flashproms.

Het probleem zit niet in de moderne of ouderwetse processor. Het gaat over de doelgroep. Binnen onze vereniging is een grote groep mensen die niet het voorrecht hebben gehad om met een Z80 op een paar eurokaartjes zelf een Tandy TRS80 te maken, compleet met disk-drives en harddisk, zelfs voor Tandy die uitbracht, maar nou praat ik over een ander probleem. Kortom, hoe laat je een (toch grote) groep geïnteresseerden zien

hoe een computer functioneert. Wij hebben gemeend dat we met een eenvoudige recht-toe-recht-aan-processor dat beter konden oplossen. Daarbij is inderdaad gedacht aan de Z80 en de opvolgers, de Z280-reeks. Omdat daarbij nogal wat hardware op de print moest komen, hebben we een goed werkend controllertje dat voor een specifieke toepassing in elkaar is gezet, van stal gehaald met duidelijk positief resultaat. In het nieuwe ontwerp voor de print van de controller is voorzien in een PIC- controller die samen met (nou ja, naast) de Intel-processor moet gaan werken. De losse bos draadjes op de tafel hiernaast, de samensmelting van de RISC-processor met de Intel 80C535 doet het aardig met B+. Trouwens, ook de mensen die dit probleem aandroegen, hartelijk dank voor de reactie, ook al vind ik dat je niet meteen al te negatief moet reageren, als je iets beter weet.



## huisbeveiliging

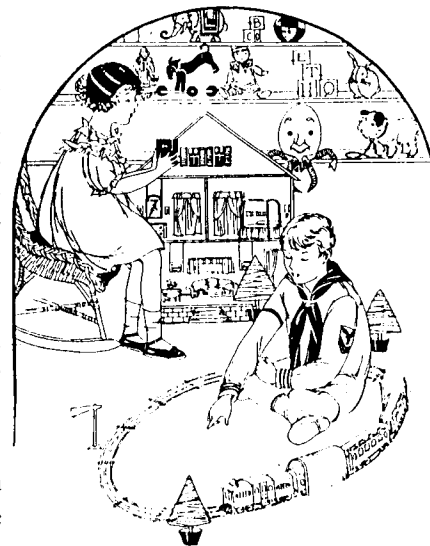
De huisbeveiliging, waar in het vorige artikel over is gesproken, werkt ondertussen feilloos. In grote lijnen zit het zo in elkaar: in twee hoeken van de kamer zitten de infrarooddetectors, heel goedkope dingen die je zo bij iedere doe-het-zelf-winkel kunt halen. Er zijn twee detectors gebruikt, omdat de ruimte niet door een detector bestreken kan worden. Als de bewegingsdetector signaal geeft, wordt er intern een relais bekrachtigd. Dat relais zorgt voor de spanning op een IC-tje dat een infrarood LED stuurt met een vast patroon van 38 kHz. Dat is precies de frequentie, waarop de afstandsbediening werkt. Dat IC-tje en die IR-LED zitten niet in de detector als u hem koopt, dat moet u er even inzetten. Mogelijk worden daar kleine printjes voor gemaakt, als er een beetje belangstelling is voor het project.

De detector stuurt geen informatie, enkel de draaggolf wordt uitgezonden. De B+-computer ziet, welk van de twee (of eventueel meerdere) detectors er actief is en kan daar op reageren.

De infraroodontvanger voor B+ stelt al helemaal niets voor. Verschillende fabrikanten (b.v. Siemens) maken een chip, speciaal voor dit soort toepassingen, drie pootjes: plus, min en uitgang. Die uitgang, die 1 wordt als er signaal wordt ontvangen, gaat rechtstreeks op een input van B+. Er wordt dus niets afgeregeld of ingesteld.

Op een verdekte plaats staat een klein kastje met daarin de B+-computer. In het kastje zijn twee gaatjes waardoor B+ de detectors kan zien. Door een derde gaatje kan B+ de (tv-) afstandsbediening zien. Via die afstandsbediening wordt het alarm aan- en uitgezet. Er kunnen ook allerlei testsignalen worden gestuurd.

B+ kan alarm maken via een geluidssignaal, of een vooraf (via de afstandsbediening) opgegeven telefoonnummer bellen. Dat bellen is erg simpel opgelost, door een 2400 baud modem (waarde  $f$  ??) aan de seriële uitgang van B+ aan de telefoon te hangen. B+ zendt een string naar het modem "ATDT" gevolgd door het nummer dat u wilt laten draaien.



De detectors kunnen worden voorzien van een nicad-accuutje als u bang bent dat de stroom uitvalt; er wordt (in beide testdetectors) slechts 12 mA opgenomen in rust en 300 mA als de IR-LED staat te stralen.

Het aardige van de hele opzet is, dat er gaan draden aan te pas komen. Niet aan alle raam- en deurcontacten en niet tussen de detectors en de computer.

Het hele programma van de beveiligingscomputer is 46 regels! Inclusief de teksten voor de testfuncties en de telefoonnummers. In de geheugenuitbreiding van B+ is ook een opstelling met een flashprom uitgetest. Deze prom moet de vervanging worden voor de RAM met batterij waarin nu het programma is opgeslagen.

## weerstanden meten

Een tweede aardige toepassing die deze week is getest, is het meten van weerstanden als functie van de tijd. Normaal zou u een onbekende weerstand aansluiten via een gekende weerstand, en dan met een van de 8 voltmeters de spanning op het knooppunt meten.

Nu hebben we de onbekende weerstand (via een stopweerstand) aan de plus gehangen en de andere kant aan een condensator. Over de condensator staat een transistor die door B+ gestuurd kan worden en dan de condensator ontlad. De condensator ligt aan een ingang van B+ (via een beschermcircuitje).

De meting gaat als volgt: B+ geeft een puls voor de transistor, de condensator is leeg en een van de timers wordt gestart. De ingang van de timer is (altijd) de systeemklok, dus  $1 \mu\text{sec}$ . B+ kijkt nu of de spanning op de condensator al 1 is. Als dat niet zo is, wordt



er nogmaals getest, enzovoort totdat de ingang wel 1 is. B+ stopt de timer en gebruikt de inhoud van de timer om de waarde van de weerstand te berekenen.

Als de onbekende weerstand een *temperatuurweerstand* is, kan B+ uit de waarde van de timer de temperatuur (of de verandering van de temperatuur) meten.

Een variant op deze meetschakeling ontstaat door de condensator te verbinden met een interruptingang. B+ blijft nu bezig met een of ander programma. Zodra het interrupt komt, wordt de condensator ontladen en gaat B+ verder met het oorspronkelijke programma. Zo is er dus een RC-oscillator (een relaxatieoscillator?) gemaakt.

Tot een volgende keer

Ton Goossens  
(079-3310893  
a.u.b. na 18.00 uur)



De belangstelling voor B+ op de themadag Programmeertalen in de Jaarbeurs was buiten alle verwachting erg goed. Niet Delphi (naast ons) had de meeste belangstelling, nee, B+ trok op de een of andere manier de meeste kijkers. Dat komt natuurlijk omdat de mensen van de NewBrain-gebruikersgroep met allerlei bewegende dingetjes zoals (hij)s kranen, robots en meer van dat soort zaken druk aan de weg timmerden. Op die themadag hebben we ook de primeur gehad van een B+ omgeving die onder Windows draait. Daar was ook erg veel belangstelling voor, zelfs van de Delphi-mensen. Het programma is geschreven in Delphi, vandaar dat onze buurman, Borland dus, luidkeels reclame maakte voor onze presentatie. Zo moet het ook.

# breadboard

Veel hobbyisten hebben een *breadboard*, nee niet moeders broodplankje, maar een testbordje, waarop je chips en andere componenten kunt prikken. Met draadjes kunnen verbindingen gemaakt worden. Heeft u nog geen

breadboardje? Gauw eentje kopen, je zult er geen spijt van hebben.

Het probleem: je moet een aansluiting maken tussen B+ en dat testbordje. Omdat je chips netjes op dat bord kunt prikken, is de afstand tussen de pennen bekend: dat is precies de afstand tussen twee gaatjes van een *montaprint* of gaatjesprint, of hoe je een dergelijk stukje eilandjesprint ook wilt noemen.

Men neme een stukje van dat eilandjesprint, en plundert uit moeders naaikastje een handje naalden. Soldeer die op dat printje. De andere zijde middels een stukje flatcable en een stekertje op de B+ aansluiten. Nu heb je een handige stekker, waarmee je het breadboard kunt aansluiten op B+.

Tekeningetje? Nee toch, dat kun je zelf toch wel bedenken! Tien naalden op een rij, en aan de andere kant een flatcable . . . Door de eenvoud heb ik deze oplossing over het hoofd gezien. Mogelijk is een ander dit ook overkomen, vandaar dit bericht.

Dré Jansen

# lcd-aansturing

Als uitbreiding op het B+-bordje is een van de mogelijkheden een LCD-schermpje. Deze scherpjes zijn er in vele soorten en maten. De bekendste zijn de scherpjes van 1 regel met 16 karakters en de scherpjes van 2 regels met 16 karakters. Maar er zijn ook scherpjes 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 scherpjes van  $2 \times 16$  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.

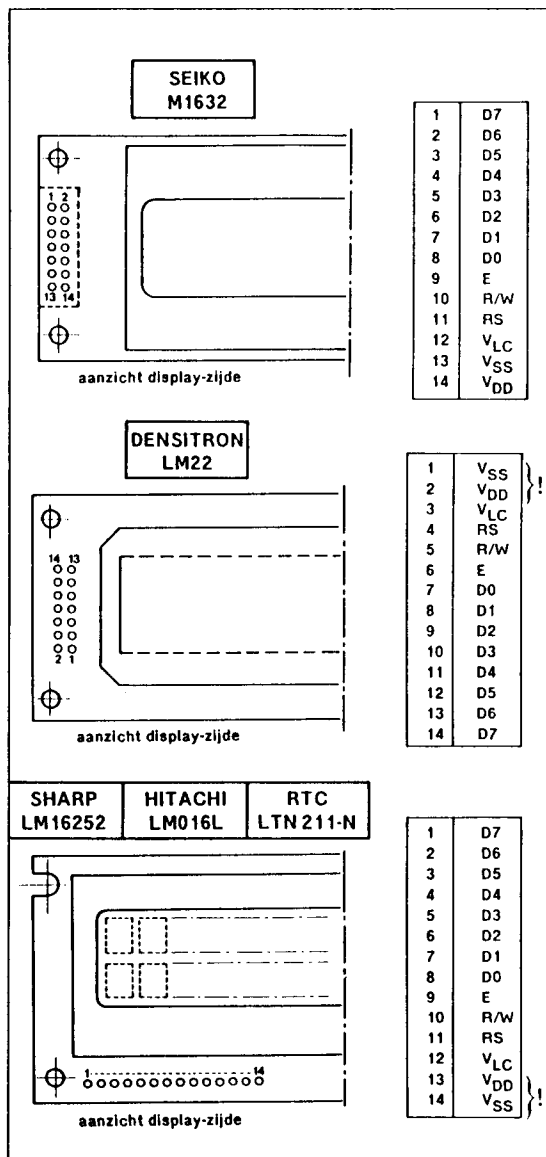


In principe werken de scherpjes als volgt: het scherpje bevat een klein geheugen, dat op het scherm afgedrukt wordt. Op de plaats waar de cursor staat, verschijnt de tekst. Op de 8 bits databus wordt de ascii-waarde van een getal of letter gezet, dan maak je de Enable hoog en weer laag en dan staat die 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 databaansluitingen, Enable, RS (Register Select), R/W (Read / Write), 5 volt, massa en de schermintensiteitsaansluiting. Daarnaast zijn er drie verschillende type schermpjes met elk hun eigen manier van aansluiten. De verschillende aansluitingen zijn in figuur 1 afgebeeld.

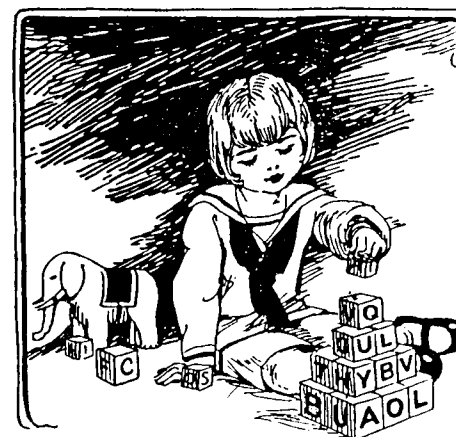
Nu is de vraag, welke schermaansluiting correspondeert met het schermje dat je in je handen hebt. Ten eerste zoek je met een ohmmeter de massa-



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 (V<sub>lc</sub>) moet je een spanning zetten tussen 0 en 5 volt. Neem dus een potmeter van 5 kΩ, zet de lopen op de V<sub>lc</sub> 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 V<sub>lc</sub> 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 schermje 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 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.





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
```

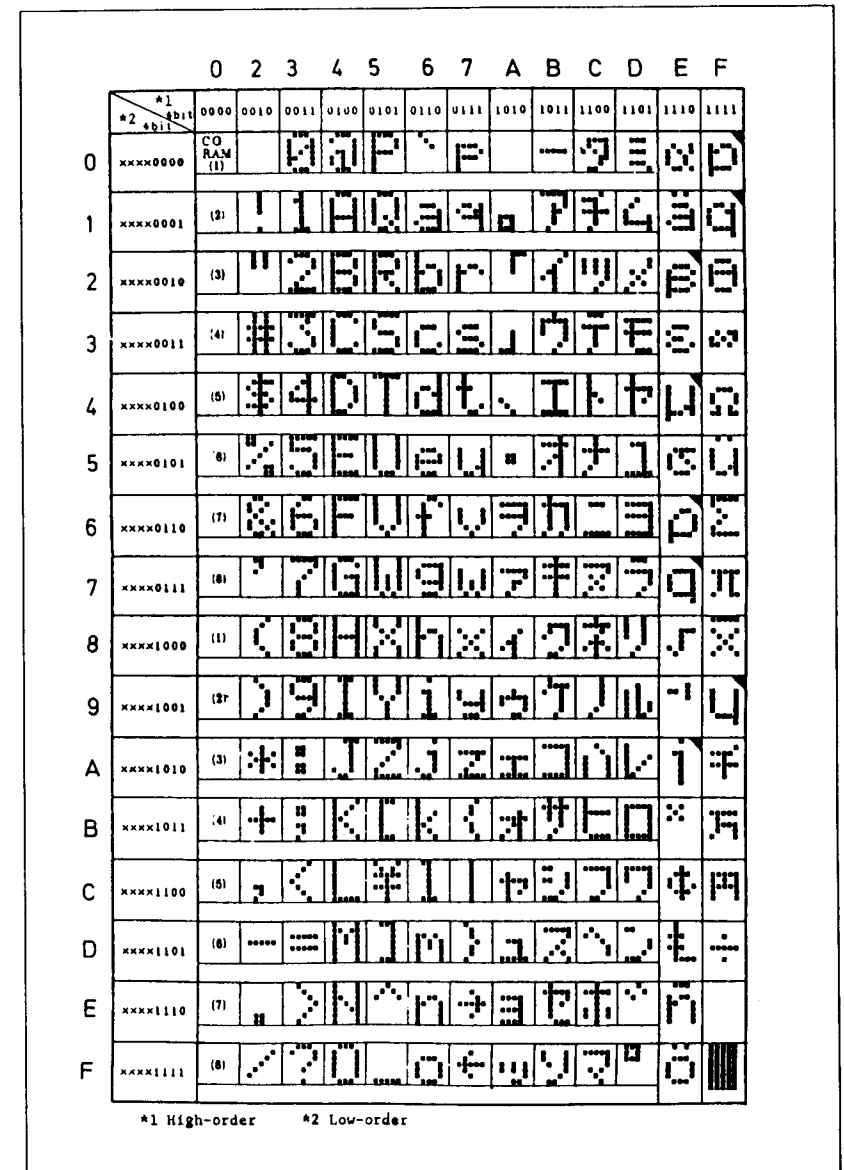
```
L1 M10=K1
,!C2
```

```
L2 M10=K2
,!C2
```

```
L3 M10=K3
,!C2
,!L2
```

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

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



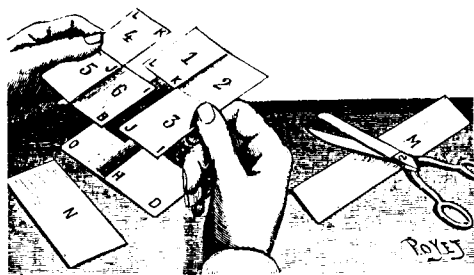
figuur 2

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

do'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	RW	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 00 (linksboven).	
cursor home	0	0	0	0	0	0	0	0	0	0	1	x	Zet de cursor op home-positie. Verschuiven van de tekst worden ongedaan gemaakt (adres 00 is weer linksboven).
verschuif tekst en/of cursor	0	0	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	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					Breedte data-bus (DL) 1 of 2 regels gebruiken (N)
karaktergenerator adres	0	0	0	0	1								Geeft aan welke rij (000...111) van welk karakter (000...111) gedefinieerd moet worden met het volgende data-byte.
geheugenadres	0	0	0	1			adres						Zet de adresssteller 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.

figuur 3

x = don't care

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 schermje 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

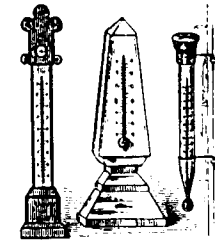
deze adressen gelden alleen, als het scherm niet verschoven is

Er zijn nog veel meer mogelijkheden met het LCD-schermje. 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 en 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
```

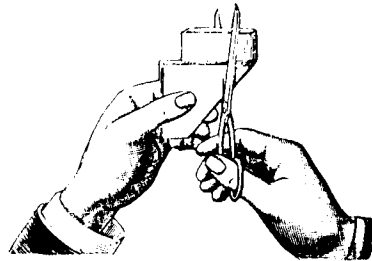
```
L7
P4=V2 .. Q50+ .. Q50- ,!C2
P4=V4 .. Q50+ .. Q50- ,!C2
P4=V5 .. Q50+ .. Q50- ,!C2
```

```
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:
```

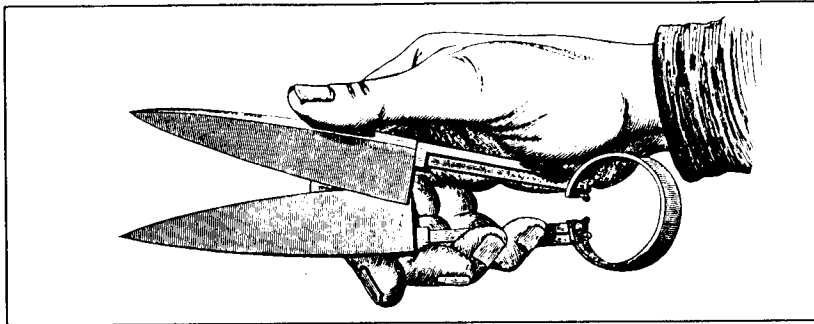
```
; P4 = DATA
; Q50 = ENABLE
; Q51 = RS
; Q52 = R/W
```

```
; 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

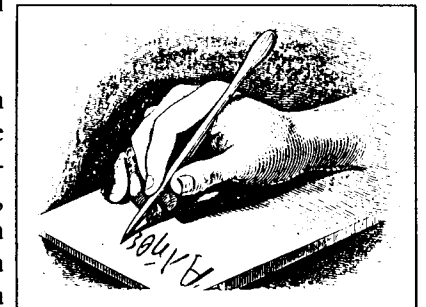


# karakter

## 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 tekenje, 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  $5 \times 8$  blokjes, en dan zetten we de hexadecimale waarde achter elk regelje. 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.

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 scherm te 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,D1,01,01,00,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

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



L2

X

In plaats van zelf 8 karakters te definiëren 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  $5 \times 8$  geprogrammeerd hebben in plaats van  $5 \times 7$ . Dat kan dus ook. Als we namelijk de cursor uitzetten, hebben we ook de beschikking over het onderste 8e

regeltje.

Succes een ieder met zijn LCD-scherm.

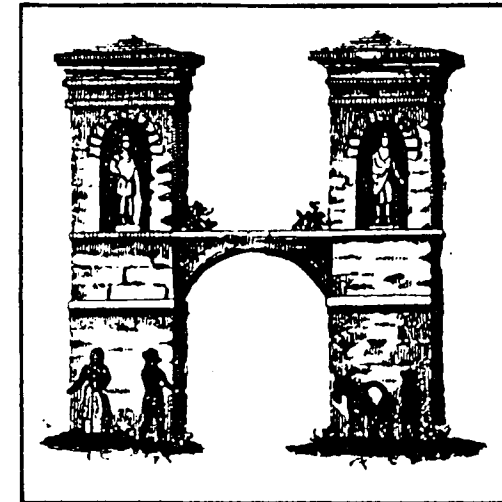
A. Vreugdenhil

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

\* don't care



# h-brug



Wanneer je een gelijkstroommotor aansluit (figuur 1-a), dan gaat hij draaien, da's logisch. Wanneer je de stroom de andere richting op laat lopen (figuur 1-b), dan draait de motor ook de andere kant op, da's iets dat niet iedereen weet. Toch is het zo. Door deze schakeling te combineren (figuur 1-c) kun je de motor beide kanten op laten lopen.

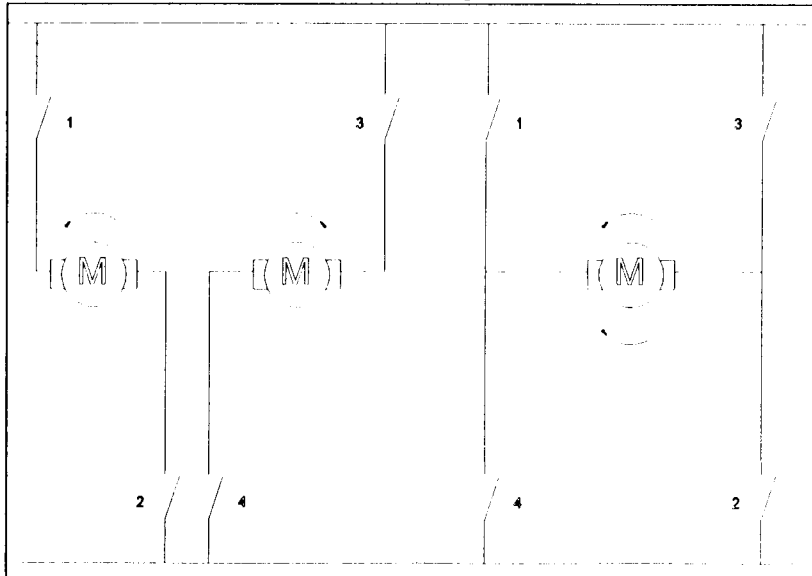
Enige voorzichtigheid is nu wel geboden, want wanneer je de schakelaars 1 en 4 gelijktijdig inschakelt, dan is er een volle kortsluiting. Zo ook met de schakelaars 3 en 2. *Niet doen dus!*

Door meer of minder spanning op de motor te zetten, kun je de draaisnelheid, ofwel het toerental regelen, dat is voor velen begrijpelijk. Een andere

manier om het toerental te regelen, is snel in- en uitschakelen. Da's moeilijker te realiseren.

In het eerste geval, is de *spanning* variabel, dus ook het koppel. In het tweede geval is de spanning constant, maar de *tijd* dat de motor ingeschakeld is, is variabel. Daarmee is het koppel eveneens constant (koppel is de motorkracht). Dus: door de schakelaar langer of korter in te schakelen, kun je, bij gelijkblijvend koppel, het toerental variëren.

Dat is mooi, heel mooi. Omdat het koppel gelijk blijft, kun je ook bij lage toerentallen de motor normaal belasten. Zou je een belaste motor met een lagere spanning langzaam willen laten draaien, dan verbrandt hij, omdat hij niet zal aanlopen. Daar zit niemand op te wachten.

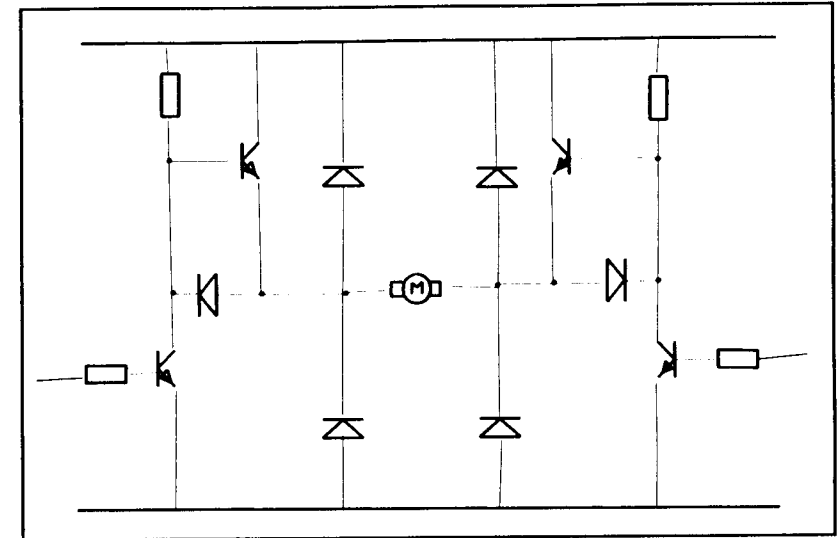


figuur 1

Natuurlijk begrijpt u, dat het onmogelijk is om met de hand de schakelaar

te bedienen. Zolang het alleen maar gaat om de richting te veranderen, is het nog wel handmatig te doen, maar met het handje de snelheid te regelen door snel in en uit te schakelen, is onmogelijk.

De schakelaar moet toch wel zo'n 100 maal per seconde worden bediend. Daar komt de elektronica om de hoek kijken.



figuur 2

Precies hetzelfde is in figuur 2 weergegeven. De schakelaars zijn vervangen door transistors. Wat meteen opvalt, is dat er wat meer onderdelen dan alleen maar vier transistors zijn getekend.

Rondom de motor zitten vier dioden. Die dienen ervoor om de inductieklappen van de motor te blussen. De transistoren kunnen daar niet tegen. Dan zitten er nog dioden tussen de transistoren. Die zijn voor het afvangen van de kortstondige kortsluiting bij het omschakelen. Door die halve volt, die over de diode staat, wordt enige schakeltraagheid gecreëerd.

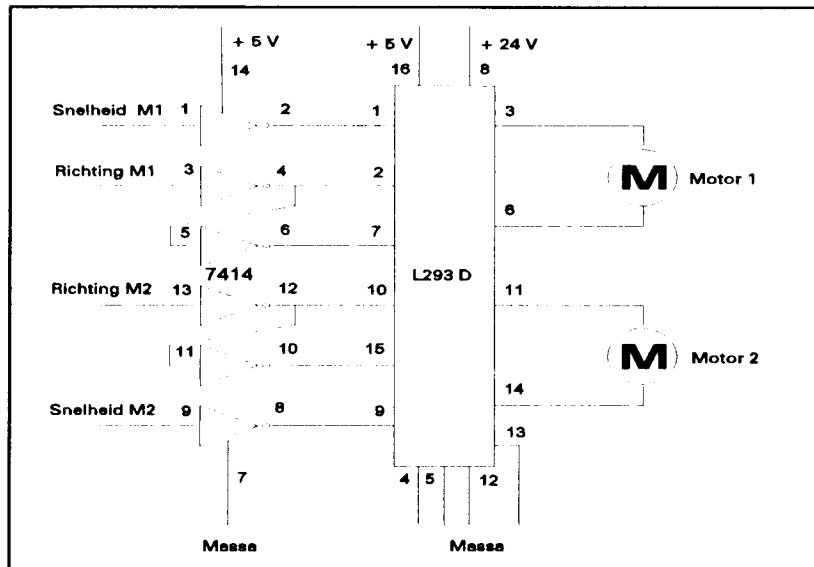


Door de bovenste transistoren te schakelen met de onderste, wordt voorkomen dat de beide 'schakelaars' gelijktijdig open staan, waardoor de gevreesde volle kortsluiting zou ontstaan.

Als je op de ene aansluiting een hoog signaal stuurt, en op de andere een laag signaal, dan loopt de motor de ene kant op; wissel je deze polariteit, dan loopt de motor andersom.

Is de polariteit op beide punten gelijk, dan staat de motor stil. U voelt het zeker al? Door enerzijds de richting te kiezen, en aan de andere kant te schakelen, kun je met twee pootjes zowel richting als snelheid bepalen.

Deze schakeling is ook verkrijgbaar in een chipje, waardoor alles nog veel eenvoudiger wordt. Nadeel is het beperkte vermogen. Chipjes kennen hun grenzen.



figuur 3



Een enkel chipje bevat de complete elektronica om twee motoren te sturen!

Wilt u ook een soldeerbout warm stoken? Dan is het prettig te weten dat er een club is met meer van die mensen. Bij ontstane moeilijkheden komen we er in clubverband wel uit.

### snelheidsregeling (pulsbreedtemodulatie)

De snelheid wordt als volgt geregeld. Men schakelt 100 maal per seconde. Deze schakelfrequentie blijft gelijk. Wat wel verandert, is de *duty cycle*. Dat is de tijd de schakelaar aan is, in verhouding tot de tijd dat de schakelaar uit is. Door die te variëren varieer je de motorsnelheid.

Wanneer je 100 maal per seconde schakelt, duurt elke schakeling 10 ms (milliseconde). Als je dan 1 ms aan en 9 ms uit zet, dan draait de motor langzaam. Doe je juist 9 ms aan en 1 ms uit, dan loopt de motor snel. Als je eens luistert naar een elektrische rolstoel, dan is deze schakelfrequentie duidelijk hoorbaar. Toch kan de rolstoel in snelheid variëren.

De frequentie is niet zo belangrijk, maar in de praktijk blijkt 100 Hz prima te voldoen.

Dré Jansen

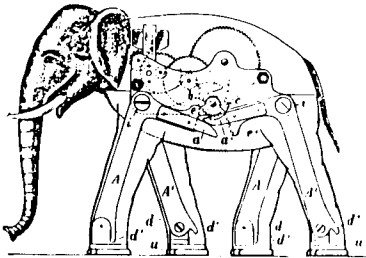




de darlington's aan massa liggen. + op 9 en - op 8. Intern zijn alle uitgangen via blusdioden met punt 9 doorverbonden.

De aansluitingen zijn dan:

+	A14	B9	motor gemeenschappelijke ader (led-weerstand zijde 1)
-	A7	B8	
data 0	A1		
data 1	A3		
data 2	A5		
data 3	A9		
	A2	B1	
	A4	B2	
	A6	B3	
	A9	B4	
motor 1	B16		(kathode ledje 1)
motor 2	B15		(kathode ledje 2)
motor 3	B14		(kathode ledje 3)
motor 4	B13		(kathode ledje 4)



(weerstand zijde 2: anode led 1, anode led 2, anode led 3, anode led 4)

Eventueel kun je nog een controleledje aansluiten. De aansluitingen zijn tussen haakjes aangegeven. Dit is vooral handig als je in de testfase bent.

*Hoe vind je de algemene motordraad?* Wanneer je met een gewone ohmmeter de weerstand tussen twee draden meet, dan meet je telkens een bepaalde waarde. Een van die waarden is ongeveer de helft van de andere waarden. Dan heb je de gemeenschappelijke draad te pakken. Welke van de twee is dan de algemene? Wel, effe uitproberen. Wanneer je telkens de lage waarde meet, dan heb je de algemene draad te pakken. Die dient aan de + gehangen te worden.

*Hoe kom je achter de juiste volgorde?* Uitproberen. Er is maar een volgorde waarbij de motor lekker draait. Sluit de motordraden willekeurig aan. Draait de motor, dan heb je mazzel, want dan heb je de goede volgorde. Anders twee draden omwisselen. Enzovoort. Hier kunnen die optionele ledjes erg handig zijn!

*Hoe laat je de motor draaien?* Welnu, daar is wat software voor nodig. Stel, je sluit de motor aan op poort 5 van B+. Als je een 1 stuurt, dan branden er 3 ledjes! Dat komt doordat je er een *inverter* tussen hebt gezet. Da's jammer. Om toch een enkele spoel te bekrachtigen, moet je een E sturen. De volgende spoel heeft D, de daaropvolgende een B en de laatste wordt met 7 aangestuurd.

Software-o-logen mogen nu prachtige programma's schrijven. Ik laat een eenvoudig programmaatje hieronder zien.

```

X                ; zoals elk B+-programma, begint ook dit met X
C1 V11= V1 V12=10          ; timervariabelen (tijd tussen 2 stappen)
  V11-01,## V12-01,##      ; de eigenlijke tijdlus
, !C0                      ; return

B                ; begin van het programma
L1                ; linksom single step
P5=E              ; de eerste stap
, !C1             ; effe wachten (subroutinecall 1)
P5=D              ; de tweede stap
, !C1
P5=B              ; derde stap
, !C1
P5=7              ; vierde stap
, !C1
V2-4, #L1        ; aantal stappen (256)

```

L2 ; *rechtsom single step*  
 P5=7 ; zoals u ziet, is deze reeks precies  
 ,!C1 ; tegengesteld aan de vorige sessie.  
 P5=B  
 ,!C1  
 P5=D  
 ,!C1  
 P5=E  
 ,!C1  
 V2-4,#L2 ; ook hier 256 stappen, en dan verder.

L3 ; *rechtsom microstepping*

**microstepping**

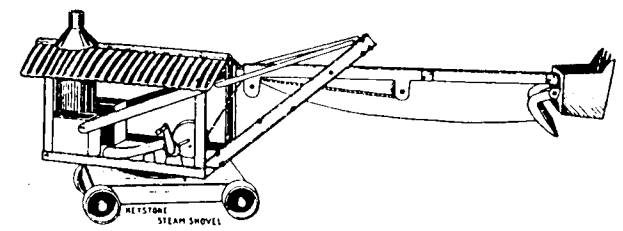
Daar hebben we het nog niet over gehad. Als je niet een maar twee spoelen bekrachtigt, dan gaat het anker precies tussen die twee magneten in staan. Daarmee is dan een halve stap gedaan. Dit noemt men *microstepping*.

Telkens wordt de opvolgende spoel bekrachtigd, maar de voorgaande blijft eveneens bekrachtigd. Pas daarna wordt de voorgaande spoel spanningsloos gemaakt. Dus afwisselend zijn er twee en een motorspoelen bekrachtigd.

De motor draait dan wel wat langzamer, maar is nauwkeuriger te positioneren.

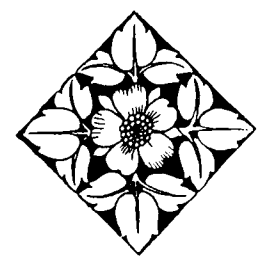
L4 ; *linksom microstepping*  
 P5=6  
 ,!C1

P5=7  
 ,!C1  
 P5=3  
 ,!C1  
 P5=B  
 ,!C1  
 P5=9  
 ,!C1  
 P5=D  
 ,!C1  
 P5=C  
 ,!C1  
 P5=E  
 ,!C1  
 V2-4,#L4 ; na 512 halve stapjes weer naar het begin  
 ,!L1 ; overnieuw  
 Y ; einde programma, Y doet meteen starten



*Hoe herken je een unipolaire stappenmotor? Draadjes tellen. Als je tot 5 kunt tellen, dan heb je een unipolaire stappenmotor. Tel je slechts 4 draadjes, geen nood, dan heb je een bipolair exemplaar. Daarover praat ik in een volgend stukje. Eerst moet ik het testprintje bouwen, dan weet ik zeker dat ik niet sta te liegen.*

Dré Jansen



# rekenen

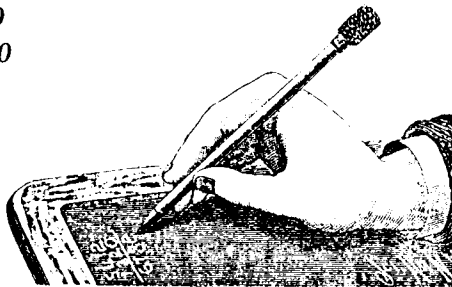
Op zaterdag 18 oktober 1997 vertelde Ton Goossens op de NewBrain-dag in Leiden een verhaaltje over *servobesturing* met B+. Later, op 1 november, herhaalde hij dit verhaal in Gouda tijdens de bijeenkomst van de Robotica-gg.

B+ verandert snel, zeer snel; was er in Leiden nog slechts sprake van conversie van hex naar dec en terug, nu was er al een complete rekenmachine geïntegreerd.

Er is een nieuwe eprom die de instructie CA heeft. Hiermee kun je getallen van hexadecimaal naar decimaal omzetten. En terug natuurlijk.

CA 20000 <enter> geeft 4E20  
CA 4E20 <enter> geeft 20000

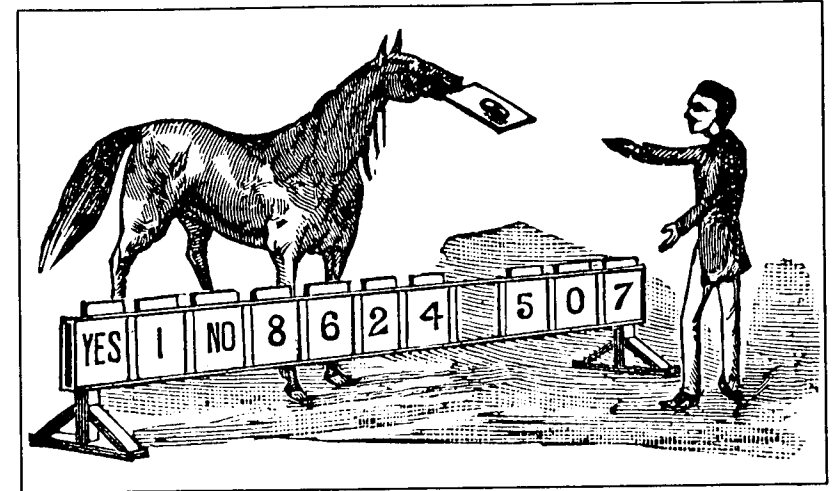
Tot zo ver duidelijk, maar als je een getal weergeeft, waarvan niet te zien is of het decimaal dan wel hexadecimaal is, moet je dat wel meegeven met de kleine letter *h* of *d*, dus CA 1234*h* geeft een decimale uitkomst, omdat er een *h* bij staat: kennelijk is 1234 hier een hexadecimaal getal.



Tot zo ver het verhaal in Leiden, in Gouda ging het verder.

Er zijn twee accu's waarmee B+ rekt, 'X' en 'Y'. Wanneer er een berekening gedaan wordt, komt het resultaat in accu 'X'. Deze accu's of registers zijn 4 bytes breed, ofwel FFFF.

Er is geen controle op *overflow*. Mocht dit van belang zijn, dan kan dat simpel worden gecontroleerd. Maak van het grote getal een hex-waarde, daarna weer decimaal, of juist andersom. Krijg je na terugkeer niet het oorspronkelijke getal terug, dan was er sprake van overflow.



Met het commando X verwissel je de inhoudswaarden van de accuregisters. X wordt Y en Y wordt X.

Het commando + telt de waarden op; resultaat komt in X

Het commando - doet juist het tegenovergestelde.

Verder gaat het rekenen (nog) niet, maar daar kan ondertussen al lang verandering in zijn aangebracht.

Dré Jansen

# inhoud

## on-line 22



- 3 newbraindag in gouda /menno stevens/
- 5 de microcontroller, deel 4 /ton goossens/
- 10 breadboard /dré jansen/
- 11 lcd-aansturing /a vreugdenhil/
- 19 a/d-omzetting naar lcd-scherm /a vreugdenhil/
- 23 eigen karakters definiëren voor een lcd-scherm /a vreugdenhil/
- 27 h-brug /dré jansen/
- 32 stappenmotor, deel 1  
aansturing van de unipolaire stappenmotor /dré jansen/
- 38 rekenen (uitbreiding van b+) /dré jansen/