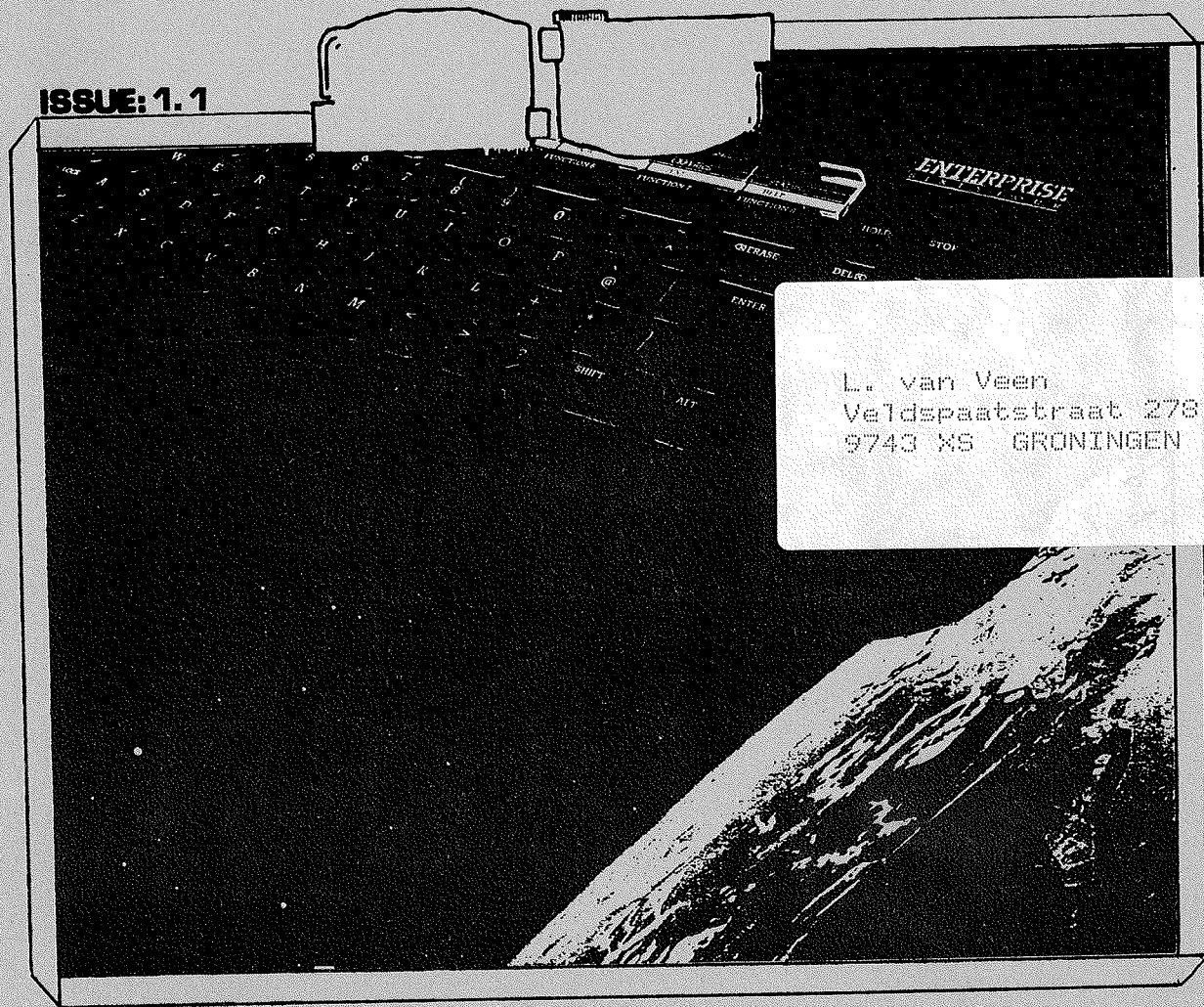


Status Regel

Port betaald
Port payé
Enschede

THE INTERFACE

ISSUE: 1.1



L. van Veen
Veldspaatstraat 278
9743 XS GRONINGEN

Mei '86

Enschede, 27-5-1986

Geachte lezers,

Voor U ligt weer een nieuwe aflevering van de INTERFACE. Nog dunner dunner dan de vorige keer, omdat er nog steeds zeer weinig copy binnen druppelt en de engelsen ook niet zo productief zijn geweest. Wij hopen dat U, zelf een bijdrage wilt gaan leveren voor de volgende edities. Dan zijn wij in staat om het blad wat vaker uit te brengen. Copy kan worden gestuurd naar:

Redactie INTERFACE
p/a Menno Bins
Dollardstr 92
7523 GS Enschede

Tevens nog maar weer eens het verzoek van de penningmeester om met spoed uw kontributie over te maken. De kontributie bedraagt f45.- per kalenderjaar en kan worden ovrgemaakt op:

giro: 5689669
t.n.v. Dutch Enterprise User Group
(Penningmeester)
De Bilt.

Gebruikersdag 26-4-1986

Alhoewel de gebruikersdag toch wel bijzonder kort van te voren was aangekondigd, bleek uit de spontane opkomst dat de belangstelling voor een ontmoeting met andere gebruikers zeer bemoedigend was. Er waren enkele noviteiten te zien zoals de bibliotheek, de muis, prototype van een backboard en wilde plannen voor verdere geheugen uitbreiding.

Op het software-front was het voor bijna iedereen overduidelijk dat Enterprise-software moeilijk verkrijgbaar is. Door navraag bij deze en gene blijkt dat men voor deze machine heeft gekozen, niet vanwege de hoeveelheid software, maar om de bijzondere eigenschappen van deze computer. Het is opvallend dat sinds de eerste gebruikersdag, de vraag naar grafische programma's, die optimaal van NICK gebruik maken, erg groot is geweest. Het aanbod voor de gebruiker, meestal een machinetaal-versie van een reeds bestaand program, komt zelden met een multi-channel video-display op dit niveau zodat de vraag blijft bestaan. Daar tegenover staat dat menig Enterprise bezitter zelf aan het programmeren is begonnen, danwel programma's voor een ander computer systeem voor de Enterprise heeft vertaald. Het is een realiteit dat 90% van de door hen gebruikte tricks van een ander zijn geleerd. De eerlijkheid gebiedt dan ook te erkennen dat de programma's vele (geestelijke) vaders hebben (sorry, dames!). In dit kader wil ik daarom met nadruk wijzen op het belang van het uitwisselen van ervaringen, tricks en oplossingen voor specifieke problemen voortspruitend uit het operating system. U hoeft niet bang te zijn dat Uw trick zo blijft, nee, U kunt er bijna gif op innemen dat mede-gebruikers, vanuit hun positie een verbeteringen weten aan te brengen. Deed U niet hetzelfde met andermans routines?? Daarom; parkeer Uw trick voor enige maanden in de bibliotheek, deponeer de door U verbeterde versie van andere tricks en zoek daar ook de routine die U als zo lang aan het uitvinden bent.

Vanuit Engeland is de eerste muis het Kanaal overgestoken. In een beschaafd antraciet grijs costuum is hij door Martin de Groot gedemonstreerd. Hij wordt aangesloten op een van de extra joystick poorten en als gevolg daarvan zijn de functies gelijk aan die van een externe joystick. De bediening is, zoals vele andere muizen, prettig en met een hand uit te voeren. Het behoeft, hoop ik, geen betoog dat programma's wel om input van de muis moeten vragen middels JOY(1) of JOY(2).

Onder de pioniers van de eerste uren heerst een grote ergernis over het uitblijven van de bestelde geheugen uitbreiding. Waar de schuld ook ligt, van deze mensen wordt erg veel geduld gevraagd. Enkele hardware specialisten binnen de gebruikersgroep bezinnen zich op een oplossing die het wachten waard lijkt te zijn. Uit een andere uithoek komt het bericht dat een backboard wordt ontwikkeld met een gebufferde tussenkaart. Op het backboard zouden dan de interfaces kunnen worden gestoken die dan voorhanden zijn. Ook op dit terrein wordt duidelijk dat we alleen met elkaar in staat zijn om de machine zodanig uit te bouwen dat het beest onder het toetsenbord de ruimte krijgt om juist die eigenschappen te vertonen waarom we hem hebben aangeschaft.

Jaarvergadering op 21 juni 1986
=====

Plaats: H.F. Witte Dorpshuis
===== Henri Dunantplein 4
3731 CL DE BILT

Agenda
=====

1. Opening en welkomstwoord.
2. Verkiezing bestuursleden.
Zie bijlage kandidaatstelling.
3. Financiële situatie.
Verantwoording over de afgelopen periode
en huidige situatie.
Instellen kascommissie.
4. Instelling werkgroepen.
 - Hardware divisie.
onder te verdelen in o.a.:
 - memory devices
 - communicatie (modem)
 - video devices
 - Software divisie.
onder te verdelen in o.a.:
 - operating systems (EXDOS & CP/M)
 - systeem-software (div. drivers)
 - educatieve software
 - business software
 - talen (Basic, Pascal, Lisp etc.)
5. Break (koffie !).
6. Instelling regionale bijeenkomsten.
7. Rondvraag.
8. Agendering volgende bijeenkomst.
9. Sluiting.

Bijlage kandidaatstelling:
Huidige kandidaten:

M. Bins
P. Degeling
B. de Kruyff
S. Tuinder

Verdere kandidaatstelling vanuit de leden dient schriftelijk te
geschieden, uiterlijk 14 juni 1986. Richt deze kandidaatstelling
aan de heer S. Tuinder
Willemstraat 170
2713 AJ ZOETERMEER.

ALGEMENE INFORMATIE

=====

- Uw bijdrage en/of bestellijst kunt u opsturen naar de Software Bibliotheek, of meenemen naar een van de algemene gebruikers bijeenkomsten die in midden Nederland gehouden worden.
- Indien uw ingezonden cassette(s)/diskette(s) niet zijn voorzien van een retour envelop met (voldoende) porto zullen deze worden meegenomen naar een van de volgende gebruikers bijeenkomst.
- Als u iets bestelt komt het regelmatig voor dat dit door u aangepast danwel verbeterd wordt. Het zou dan prettig zijn als u deze updated file weer beschikbaar stelt. Zo'n updated file dient dan ook te zijn voorzien van uw naam en telefoonnummer NAAST die van de oorspronkelijke auteur samen met 'WAT, WAAROM en WANEER' u een verandering heeft aangebracht. Dit kan eventueel in een aparte text (.UPD) file worden vermeld.
- Als u vragen heeft over een van de programma's uit de bibliotheek, neem dan zo mogelijk contact op met de auteur van dit programma, en NIET met de Software Bibliotheek.
- Helaas heeft de bibliotheek nog niet de beschikking over een 3.5" diskdrive, en kunnen mensen met zo'n drive (voorlopig) alleen cassettes opsturen of hun eigen drive meenemen naar een van de algemene bijeenkomsten.
- Het is misschien ook verstandig om uw eigen cassette recorder mee te nemen naar de bijeenkomst, dan bent u verzekerd van een door u leesbare cassette.

Correspondentie adres: Software Bibliotheek,
 p/a Reigerskamp 690,
 3607 JP Maarssen.

Richard Sargeant.
03465-65086
(19:00 - 22:00 uur)

D.E.U.G. SOFTWARE BIBLIOTHEEK

=====

Zoals sommige onder u al weten heeft onze gebruikers groep nu een Software Bibliotheek (eindelijk!). Natuurlijk kan zo'n Bibliotheek alleen goed functioneren met uw bijdragen. Veel mensen zullen echter denken dat zij niets bij te dragen hebben, maar dat hoeft niet het geval te zijn. Iedereen heeft zeker een programma, routine of opmerking voor de Enterprise die voor anderen nuttig zou kunnen zijn.

Om het gebruik van de bibliotheek voor u en voor de beheerders te vergemakkelijken hebben wij het volgende reglement opgesteld:

HOE STUURT U UW BIJDRAGE NAAR DE BIBLIOTHEEK

=====

- allereerst dient iedere bijdrage te zijn voorzien van:

1. Uw naam en telefoonnummer.
(in de source en/of document file)
2. Indien nodig een Document/Help file met gebruiksaanwijzing.
3. Een filename - max 8 chrs. met een standaard extension.
(zie ook 'Standaard File Extensions')

- voor de bibliotheek is de volgende informatie noodzakelijk:

1. Taal/Systeem (Basic, Pascal, ISDOS enz.)
2. Korte beschrijving maximaal 40 chars. (per file)
3. Benodigdheden - 64/128k, printer, disk/cassette enz.
(zover bekend)

NB. Uw programma's worden ook bekeken door mensen met misschien wat minder ervaring dan u heeft. Het is daarom makkelijk als er voldoende commentaar bij de source file geleverd wordt.

HOE BESTELT U IETS UIT DE BIBLIOTHEEK

=====

- Uw aanvraag dient te zijn voorzien van een lege cassette(s) of GEFORMATEERDE diskette(s) met voldoende ruimte voor de door u opgevraagde files.

- Uw bestellijst moet bestaan uit 'code' en 'filename.ext' in volgorde van 'code'!

D.E.U.G. SOFTWARE LIBRARY

STANDAARD FILE EXTENSIONS

Om files makkelijk te kunnen identificeren wordt er gebruik gemaakt van standaard EXDOS filenames (max 8 characters) met een van de volgende extensions.

<i>ext</i>	<i>beschrijving</i>
.4TH	Forth file
.APP	Application
.ASC	ASCII file
.ASM	Assembler source file
.BAS	Single BASIC file
.C	C source file
.COM	ISDOS command file
.DAT	Data file
.DOC	Document file
.GEN	DEVPAC source file
.H	C header file
.HLP	Help file
.MBS	Multiple BASIC program file
.MRN	Multiple BASIC program file (autostart)
.OVR	Overlay file
.PAS	PASCAL file
.PRT	Print file
.RUN	Autostart BASIC program
.TXT	Text file
.XA	Absolute system extension
.XR	Relocatable system extension
.WP4	40 column Word Processor file
.WP8	80 column Word Processor file

*** ADVENTURES ***

code	filename	ext	bytes	language	description	received	required
A.001.01	ADVENT	DIR	194304	IS-DOS	Uitgrebreid 'Colossal Cave'	01/04/86	D/E
A.001.02	GEVAAR	BAS	22743	IS-BASIC	Nederlandstalig Adventure	13/04/86	

*** BUSINESS ***

code	filename	ext	bytes	language	description	received	required
B.001.01	NPS_CBL	DIR	57728	IS-DOS	NPS Micro Cobol ver 2.1	01/04/86	D
B.001.02	SPELL	DIR	156002	IS-DOS	Spelling checker with English dictionary	04/04/86	D/E
N> B.001.03	KRTN_BAK	DIR	17093	IS-BASIC	Kaartenbak systeem	26/04/86	C

*** DEMONSTRATIONS ***

code	filename	ext	bytes	language	description	received	required
D.001.01	NETWORK	BAS	12672	IS-BASIC	Network demo	01/02/86	
D.001.02	PICTURES	DIR	115259	IS-BASIC	Hires pictures	01/04/86	E
D.001.03	ENVELOPE	BAS	5263	IS-BASIC	Sound demo	26/04/86	

*** GAMES ***

code	filename	ext	bytes	language	description	received	required
G.001.01	Z80CHESS	DIR	13312	IS-DOS	Chess	01/04/86	D/E
N> G.001.02	OTHELLO	BAS	17201	IS-BASIC	Othello (reversi)	26/04/86	
N> G.001.03	WOOPER	BAS	16455	IS-BASIC	Spelletje met sprites	26/04/86	
G.001.04	HAPSLANG	BAS	3368	IS-BASIC	Eet de vliegen	26/04/86	
G.001.05	EXPERT	BAS	3162	IS-BASIC	De computer raad het !	26/04/86	C

*** ROUTINES ***

code	filename	ext	bytes	language	description	received	required
R.001.01	CONVERT	BAS	1780	IS-BASIC	Hex/Decimal/Binary conversion routines	01/02/86	
R.001.02	SORT	BAS	1559	IS-BASIC	Sort routine	01/02/86	
R.001.03	ALLOCATE	BAS	534	IS-BASIC	Allocate bug work-a-round	01/02/86	
R.001.04	XR_HEAD	GEN	1792	DEVPACK	Standard Header for system extensions	01/02/86	

D.E.U.G. SOFTWARE LIBRARY

*** TEXTFILES ***

code	filename	ext	bytes	language	description	received	required
T.001.01	MEM_MAP	WPB	9088	WP	Memory map 25-01-86	01/02/86	
T.001.02	BASIC_1	WPB	3712	WP	Hoe BASIC opgeslagen wordt	01/02/86	
T.001.03	CHAR_SET	WPB	4736	WP	Character set table	01/02/86	

*** UTILITIES ***

code	filename	ext	bytes	language	description	received	required
U.001.01	CO80	DIR	4470	DEVPACK	Veranderd EDITOR naar 80 columns	01/02/86	E
U.001.02	SD	DIR	2176	IS-BASIC	Sorted directory naar FX-80	01/02/86	D/P
U.001.03	DATA_GEN	BAS	2196	IS-BASIC	Data statement generator	01/02/86	
U.001.04	LU	DIR	20824	IS-DOS	Library Utility	01/04/86	D/E
U.001.05	SQUEEZE	DIR	7040	IS-DOS	File squeezer	13/04/86	D/E
U.001.06	LOCK	DIR	25472	IS-DOS	File Lock/Unlock	13/04/86	D/E
U.001.07	VSLD	DIR	3120	IS-BASIC	Video Save/Load/Dump utilities	13/04/86	
>U> U.001.08	BASIC20	DIR	17577	IS-BASIC	Memory map - IS-BASIC 2.0	26/04/86	
U.001.09	MONITOR	DIR	28890	IS-BASIC	Monitor	11/04/86	
U.001.10	DISSASM	DIR	11266	IS-BASIC	Disassembler	11/04/86	
U.001.11	FONTSY	DIR	125184	IS-DOS	Banner print utility	25/04/86	D/P/E
>N> U.001.12	KBRD	DIR	4882	DEVPACK	Keyboard translator for Wordstar	26/04/86	D/E
>N> U.001.13	SDUMP	DIR	8599	DEVPACK	Screen dump naar EPSON	26/04/86	P
>N> U.001.14	HEXDUMP	BAS	11477	IS-BASIC	Hex dump (mooie scherm indeling)	02/05/86	

EXOS Fout-codes

De hieronder afgedrukte foutcodes worden na een EXOS-call geretourneerd in register A van de Z-80. De betekenis van de foutcode staat erachter.

;Algemene foutcodes geretourneerd door EXOS-kernel :

FF *** Invalid EXOS function code
FE *** EXOS function call not allowed
FD *** Invalid EXOS string
FC *** EXOS stack overflow
FB *** Channel does not exist
FA *** Device does not exist
F9 *** Channel already exists
F8 *** Channel open error
F7 *** Insufficient memory
F6 *** Insufficient video memory
F5 *** No free segment
F4 *** Cannot free segment
F3 *** Invalid user boundary
F2 *** Unknown EXOS variable number
F1 *** Invalid device descriptor
F0 *** Unrecognised command string
EF *** Invalid enterprise file header
EE *** Unknown module type
ED *** Invalid relocatable module
EC *** End of file module
EB *** Invalid date or time value
EA *** Invalid special function call
E9 *** Device already in use
E8 *** Invalid unit number
E7 *** Call not supported by this device
E6 *** Invalid escape sequence
E5 *** STOP key pressed
E4 *** Unexpected end of file
E3 *** Protection violation
E2 *** Function key string too long

SOUND-Errors :

E1 *** Envelope too big
E0 *** Envelope storage full
DF *** Sound queue full

VIDEO-Errors :

DE *** Invalid video page size
DD *** Invalid video mode
DC *** Invalid display parameters
DB *** Invalid video page file
DA *** Invalid row number to scroll
D9 *** Invalid cursor coordinates
D8 *** Invalid beam position

Een programma in een functionele taal is veelal een set definities van functies een "aanroep" van een van die functies zorgt dat het programma gaat lopen. In LISP wordt gewerkt met S-expressies, S-expressies zijn atoms of dotted-pairs. Atoms zijn identifiers, karakters of getallen, dotted-pairs zijn de koppel-elementen van lijsten, ze worden gerepresenteerd door (<S-expr> . <S-expr>).

Bijvoorbeeld 17, A en APQR zij atoms, ((A.B).C) is een dotted pair met als eerste S-expressie (de CAR) weer een dotted pair, en als tweede (de CDR) het atom A. Lijsten worden gevormd uit dotted pairs d.m.v. de volgende recurrente betrekking:

```
Lijst ::= NIL   of
        ( S-expr . Lijst )
```

NIL wordt ook wel als () geschreven. Voor lijsten bestaat er een verkorte notatie, die ontstaat door .(en het bijbehorende) weg te laten. Bv. (A.(B.(C.NIL))) -> (A B C)

```
want .NIL = .() =
,((A.NIL).NIL) -> ((A))
,(A.((B.C).(D.NIL))) -> (A (B.C) D)
,(NIL.NIL) -> (NIL) -> (())
```

Hiermee zijn lijsten te schrijven als

```
( <S-expr1> ... <S-exprn> )
```

Programma's in LISP zijn altijd S-expressies, als ze door de LISP interpretator worden geevalueerd, wordt de eerste element van een lijst gezien als een functie die moet worden toegepast op de argumenten die de rest van de lijst vormen.

```
(PLUS X Y)   betekent dus X+Y
```

Twee belangrijke constructies in LISP zijn de conditionele expressie en de functie definitie. De konditionele expressie heeft de vorm

```
(COND ( <Cond1> <S-expr1> )
      ( <Cond2> <S-expr2> )
      ..
      ..
      ( <Condn> <S-exprn> ))
```

In BASIC zou ditzelfde er ongeveer zo uit zien

```
IF <Cond1> THEN
  <S-expr1>
ELSE IF <Cond2> THEN
  <S-expr2>
ELSE IF ...
  ..
  ..
ELSE IF <Condn> THEN
  <S-exprn>
ENDIF
```

Met dat verschil dat in LISP alles dat ongelijk aan NIL is als waar wordt geïnterpreteerd en NIL als onwaar.

Dus

```
(COND ((PLUS 1 2) (PRINT "1+2=waar"))
      (T          (PRINT "1+2=onwaar")))
```

drukt "1+2=waar" af. (T staat voor true, wat engels is voor waar)

Dan hebben we nog de functie definitie:

```
(DEFUN <Fname> <ArgList> <S-expr>)
```

<Fname> is de naam van de te definiëren functie, <ArgList> is een lijst met argumenten en de <S-expr> is de "Body" van de functie.

```
bv. (DEFUN F (X Y) (PLUS X Y))
```

definieert de functie $f(x,y)=x+y$.

Er zijn nog veel meer mogelijkheden, het voert te ver om daar hier op in te gaan, tenslotte is dit geen cursus LISP. Aan de hand van een paar voorbeelden zal ik proberen uit te leggen wat functioneel programmeren nu eigenlijk precies is, maar allereerst moet ik nog opmerken dat LISP strikt gesproken geen functionele taal is, omdat en dat is het grote verschil tussen functionele en conventionele talen, LISP-functies neven effecten kunnen geven, zoals het toekennen van eigenschappen aan atomen. Maar het is wel degelijk mogelijk om in LISP functioneel te programmeren, we moeten dan gewoon die functies die die neveneffecten opleveren niet gebruiken. Die functies zijn: SET, SETQ, PUT en strikt gesproken ook de PRINT-functies. (Er zijn er nog meer, maar die zijn minder belangrijk).

```
(DEFUN FAC (N)
  (COND ((EQUAL N 0) 1)
        (T (TIMES N (FAC (SUB1 N)))
  )
)
```

Hier wordt de functie FAC gedefinieerd als

```
FAC(n) ::= { 1 ,als n=0
            { n*FAC(n-1) ,anders
```

Dit is precies de wiskundige definitie van de faculteit. De definitie (wiskundig En in LISP) is een recurrente betrekking die fac definieert voor elke n. In het algemeen is het zo dat een wiskundige formulering van het probleem onmiddellijk tot het functionele programma leidt. Het is natuurlijk zo dat door de primitieve opzet van LISP de definities zeer slecht leesbaar zijn (veel haakjes en onconventionele notatie voor functie-aanroep), maar dat geldt niet voor alle functionele talen, in TWENTEL is het bovenstaande programmaatje:

```
DEF
  fac n = IF n=0 -> 1,
           -> n*fac(n-1)
FI
FED
```

TWENTEL is wel strikt functioneel, en kan daardoor ook met oneindig lange lijsten werken, maar misschien schrijf ik daar een andere keer een verhaaltje over.

Dr. Bill

In deze rubriek zullen wij ons bezig houden met HARDWARE uitbreidingen voor onze ENTERPRISE !!! Maar voor we de soldeerbout uit de kast halen, zullen we eerst een inzicht willen hebben in de wijze waarop we van alles aan de ENTERPRISE kunnen hangen. Allereerst worden we dan geconfronteerd met de indeling en toewijding van geheugen en I/O (Input, Output) adressen door I.S. en derden (MABAN-NEDERLAND)

----- SEGMENT ----- GEBRUIK -----

00-03	EXOS
04-07	ROM-BAY
08-20	FREE
20-2F	EXDOS
2F-6F	FREE
70-77	EPROM PROGRAMMER
78-F7	FREE
F8-FB	RAM (op de 128)
FC-FF	VIDEORAM

----- I/O ADRES ----- GEBRUIK -----

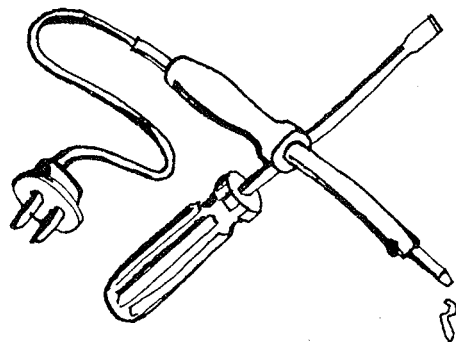
00-7B	FREE
7C-7F	EPROM PROGRAMMER
80-FF	ENTERPRISE & I.S.

Wanneer wij EXOS opstarten is het eerste wat hij (zij?) doet er proberen achter te komen in wat voor configuratie de ENTERPRISE zich bevindt. Hiertoe wordt gekeken waar zich RAM en ROM bevinden. Op gevonden RAM wordt een test gedaan om te bekijken of het naar behoren functioneert. Met ROM ligt het wat anders daar dit enkel in de 4 segmenten van de ROM-BAY en op iedere 256k. grens gezocht wordt. Zo wordt dus een EXOS-ROM in segment 0A niet als zodanig herkend. Een ROM mag echter wel niet volledig uitgedecodeerd worden en zich dus meerdere malen in een 256k. gebied bevinden. Dit gebeurt bijvoorbeeld met EXDOS dat nog wat meer eigenaardigheden bezit het kan zich namelijk door het geheugen verplaatsen wanneer een van de slot connectoren benut wordt. Standaard bevindt het zich echter op segmenten (20-2F).

We weten nu dus waar we wat kunnen aan hangen en eigenlijk houdt niets ons nu tegen om naar believen RAM, ROM en DEVICES met de ENTERPRISE te verbinden. Om onze knutsel activiteiten en die van derden echter zo te houden dat ze onderling uitwisselbaar zijn verdient het aanbeveling om nog wat meer afspraken te maken bijvoorbeeld over de bus aansluiting die wij zullen gebruiken. De uitbreidingen die in deze rubriek aan de orde zullen komen, zullen, voor zover zij eveneens als kant en klaar product worden geleverd, allen direct (wanneer zij zich buiten de ENTERPRISE bevinden) op de bus van de BACKPLAIN van (MABAN-NEDERLAND) aan gesloten kunnen worden. Het belangrijkste is echter dat wanneer men zich bijvoorbeeld een I/O adres toeigent men dit door geeft aan (nog beter overlegt met) de GEBRUIKERS-VERENIGING, zo dat deze op zijn beurt anderen op de hoogte kan stellen van het gebruik van dit adres.

Dr. Bill

Daar we weten dat niet een ieder de mogelijkheid heeft zelf een uitbreiding voor de ENTERPRISE te knutselen zullen we proberen er zelf enkele voor deze groep mensen te ontwikkelen. Zo wordt er op dit moment hard gewerkt aan een 256K (uitbreidbaar tot 512K !!!) interne geheugen kaart die tevens een omschakelbare klok (4M/6M) frequentie heeft. Een prototype valt hopelijk op de volgende gebruikersdag te bewonderen.



Maar voor de haastigen onder ons. hier alvast een mogelijk om al je programma's voor maar liefst 50% te versnellen.

Conditie: Een Enterprise met EXOS in rom (niet in EPROM, daar deze langzamer werkt) en geen diskdrive. Disk-access op verhoogde snelheid is niet zondermeer mogelijk. Misschien na wat veranderingen in EXDOS ?? Als een van jullie dit uitzoekt laat het resultaat dan s.v.p. weten.

Benodigd: Een 12 Megahertz kristal en een 56 pico-Farat condensator.

Handeling: Vervang het 8 Megahertz kristal door die van 12 MHz en condensator C7 door een van 56 p. Aan hen die normalerwijze niet met een soldeerbout elektronika bewerken nog de volgende tips.

- 1) Niet iedere soldeerbout is geschikt om elektronica componenten mee te solderen, dit heeft te maken met het geleverde vermogen en vooral met de grootte van de soldeerpunt. Gebruik dus van te voren je verstand en koop indien nodig eerst een geschikte soldeerbout.
- 2) haal de printplaat uit de ENTERPRISE en soldeer de komponenten aan de onderkant los, dit werkt over het algemeen het prettigst.

Dr. Bill

Kosten: De kosten voor dit project bedragen ongeveer 14 gulden.

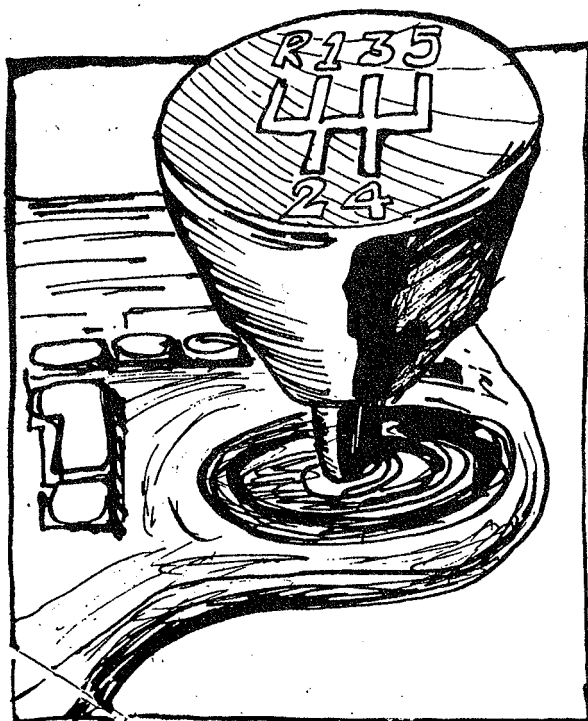
Effect: Onze ENTERPRISE heeft na deze simpele ingreep een 12 Mhz. klok en loopt dus 50% sneller. dientengevolge verricht hij de meeste handelingen ook 50% sneller maar dit is niet altijd gewenst, denk bijvoorbeeld aan het bijhouden van de juiste tijd.

Om nu de vele interrupts die door DAVE veroorzaakt worden weer met hun normale frequentie te laten komen heeft de ENTERPRISE een I/O adres waar we hem kunnen vertellen met wat voor klok we werken b.v door middel van OUT 191,xx .

Bit 2 (byte=1 t/m 8) geeft aan of het zich om ,wanneer (0), een 8MHz of een, wanneer (1), 12MHz ENTERPRISE handelt. Wanneer men bovendien bit 3 en 4 op (1), (0) zet halen we nog wat wait's uit het access, op alles wat niet in het VIDEO-RAM ligt. We hebben nu dus gekregen out 191,10 .

Veel success bij het ombouwen.

Dr. Bill .



STATUS-REGEL

=====

Veel mensen die mijn programma's aanschouwen, staan stomverbaasd te kijken hoe ik met de status-regel manipuleer. Deze mensen kan ik geruststellen omdat alles wordt verzorgd door een simpele doch adequate routine. Door deze routine in Uw eigen programma's te verwerken kunt U ook van de status-regel gebruik maken.

Dit is de routine voor de Enterprise 64:

```
1000 DEF STATUS(A$,A)
1010   POKE 47881,A:POKE 47883,A
1020   LET A$=A$&" " ! (35 SPATIES)
1030   LET POS=PEEK(49142)+256*PEEK(49143)+4
1040   FOR C=1 TO 35
1050     POKE POS+C,ORD(A$(C:C))
1060   NEXT C
1070 END DEF
```

Voor de Enterprise 128 dient U de waarden in regel 1010 te wijzigen in 47369 en 47371.

En zo roept U de routine aan:

```
10 CALL STATUS(TEKST$,KLEUR)
of
10 CALL STATUS("Dit is de tekst",123)
```

BASIC-programma's in Uw brief

=====

Het is mogelijk om heel gemakkelijk een BASIC-programma in Uw brief verwerken zonder dat U het opnieuw moet intikken. Dit intikken leidt vaak tot fouten, welke in programma's natuurlijk uit den boze zijn. Namens de redactie van het clubblad verzoek ik U zoveel mogelijk van deze truc gebruik te maken.

Het gaat als volgt in zijn werk.

- maak het programma
- geef de commando's OPEN #200:"TAPE:BASIC-WP" ACCESS OUTPUT
LIST #200
CLOSE #200

Als U de beschikking over een EXDOS-module heeft, gebruikt U natuurlijk de RAMDISK.

Deze truc is overigens niet alleen toepasbaar voor BASIC-programma's maar werkt bijvoorbeeld ook met files van GEN (assembler-deel van DEVFAC).

Stap in de WP en U kunt nu de listing laden. Over werkbeparing gesproken....

Bas de Kruijff.

Drie dimensionaal tekenen.

Hieronder staan twee programma's waarmee 3D-tekeningen kunnen worden gemaakt. Het eerste verwacht dat op disk of op cassette een file staat met informatie over het te tekenen object. Het tweede programma maakt zo'n objectfile voor een kubus. Het tekenprogramma bekijkt het object vanuit een bepaalde invalshoek, de coördinaten van het gezichtspunt staan in de variabelen o1,o2 en o3, ze worden ingesteld op regel 510. De grootte kan worden ingesteld m.b.v. de variabele SCREEN-DISTANCE in regel 520.

tek

```
100 PROGRAM "3D_draw"
101 ! Written by   Gert Veldhuijzen van Zanten
102 !             Calslaan 13-208
103 !             7522 MH Enschede
104 !
110 NUMERIC X1,Y1,Z1,O1
120 NUMERIC X2,Y2,Z2,O2
130 NUMERIC X3,Y3,Z3,O3
140 NUMERIC SCREEN_DISTANCE
150 NUMERIC NR_OF_LINES,NR_OF_POINTS
160 LET NR_OF_POINTS=50:LET NR_OF_LINES=100
170 NUMERIC PNT(1 TO NR_OF_POINTS,1 TO 3)
180 NUMERIC LNS(1 TO NR_OF_LINES,1 TO 2)
190 CALL DEFINE_OBJECT
200 GRAPHICS
210 SET STATUS OFF
220 SET #102:PALETTE WHITE,BLACK,WHITE,RED
230 SET #101:PALETTE WHITE,BLACK,GREEN,RED
240 SET BORDER WHITE
250 CALL MAIN
260 END
270 !
280 DEF MAIN
290   NUMERIC P1,P2,P3,X,Y
300   LET EYE_DISTANCE=NORM(O1,O2,O3)
310   LET Z1=-O1/EYE_DISTANCE
320   LET Z2=-O2/EYE_DISTANCE
330   LET Z3=-O3/EYE_DISTANCE
340   CALL CALC_EYE_BASE
350   FOR LINE=1 TO NR_OF_LINES
360     SET BEAM OFF
370     FOR POINT=1 TO 2
380       LET P1=PNT(LNS(LINE,POINT),1)
390       LET P2=PNT(LNS(LINE,POINT),2)
400       LET P3=PNT(LNS(LINE,POINT),3)
410       CALL COORD_TRANSF(P1,P2,P3)
420       CALL PROJECTION(P1,P2,P3,X,Y)
430       PLOT X*10+600,Y*10+300;
440     NEXT POINT
450   NEXT LINE
460 END DEF
470 !
480 DEF DEFINE_OBJECT
490   INPUT PROMPT "Give object-file :":FILE$
500   OPEN #1:FILE$
510   LET O1=20:LET O2=20:LET O3=10
520   LET SCREEN_DISTANCE=250
530   LET NR_OF_POINTS=-1
```

```

540 DO
550 LET NR_OF_POINTS=NR_OF_POINTS+1
560 FOR COORD=1 TO 3
570 INPUT #1:D$
580 IF D$="start of lines" THEN EXIT FOR
590 LET PNT(NR_OF_POINTS+1,COORD)=VAL(D$)
600 NEXT COORD
610 LOOP UNTIL D$="start of lines"
620 LET NR_OF_LINES=-1
630 DO
640 LET NR_OF_LINES=NR_OF_LINES+1
650 FOR POINT=1 TO 2
660 INPUT #1:D$
670 IF D$="end of lines" THEN EXIT FOR
680 LET LNS(NR_OF_LINES+1,POINT)=VAL(D$)
690 NEXT POINT
700 LOOP UNTIL D$="end of lines"
710 CLOSE #1
720 END DEF
730 !
740 !
750 DEF CALC_EYE_BASE
760 ! Calculate Eye_base_vectors (x1,x2,x3) and (y1,y2,y3)
770 NUMERIC A,B,C
780 IF Z2=0 THEN
790 LET Y1=-Z3*SIGN(Z1)
800 LET Y2=0
810 LET Y3=ABS(Z1)
820 ELSE IF Z3=0 THEN
830 LET Y1,Y2=0
840 LET Y3=1
850 ELSE
860 LET A=Z1/Z2
870 LET B=-(Z2*Z2+Z1*Z1)/(Z3*Z2)
880 LET C=-SGN(Z2*Z3)*NORM(A,1,B) !factor to scale |y| to 1
890 LET Y1=A/C
900 LET Y2=1/C
910 LET Y3=B/C
920 END IF
930 LET X1=Z2*Y3-Z3*Y2 ! |y|=|z|=1 => |(y*z)|=1
940 LET X2=Z3*Y1-Z1*Y3
950 LET X3=Z1*Y2-Z2*Y1
960 END DEF
970 !
980 DEF COORD_TRANSF(REF C1,REF C2,REF C3)
990 ! Uses global variables :
1000 ! Eye_coordinates : o1, o2, o3
1010 ! Sight_direction : z1, z2, z3
1020 ! Vert._screen_direction : y1, y2, y3
1030 ! Horz._screen_direction : x1, x2, x3
1040 !
1050 NUMERIC CC1,CC2,CC3
1060 LET CC1=C1:LET CC2=C2:LET CC3=C3
1070 LET C1=X1*(CC1-O1)+X2*(CC2-O2)+X3*(CC3-O3)
1080 LET C2=Y1*(CC1-O1)+Y2*(CC2-O2)+Y3*(CC3-O3)
1090 LET C3=Z1*(CC1-O1)+Z2*(CC2-O2)+Z3*(CC3-O3)
1100 END DEF
1110 !

```

```

1120 DEF PROJECTION(C1,C2,C3,REF X,REF Y)
1130 ! Uses global variable screen_distance
1140 ! denoting the distance between eye and screen
1150 NUMERIC F
1160 LET F=SCREEN_DISTANCE/C3
1170 LET X=C1*F
1180 LET Y=C2*F
1190 END DEF
1200 !
1210 DEF NORM(P,Q,R)=SQR(P*P+Q*Q+R*R)
1220 DEF SIGN(X)=SGN(SGN(X)+.5)

```

files

```

10 PROGRAM "Create_object-files"
20 ! This program creates an object file for the 3d program
30 ! also written by Gert Veldhuijzen van Zanten
40 ! Calslaan 13-208
50 ! 7522 MH Enschede
100 INPUT PROMPT "Give object-file ":FILE$
110 OPEN #1:FILE$ ACCESS OUTPUT
120 RESTORE 200
130 DO
140 READ D$
150 PRINT #1:D$
160 LOOP UNTIL D$="end of lines"
170 CLOSE #1
180 END
190 !
200 DEF DATA
210 DATA -1,-1,-1
220 DATA 1,-1,-1
230 DATA -1,1,-1
240 DATA 1,1,-1
250 DATA -1,-1,1
260 DATA 1,-1,1
270 DATA -1,1,1
280 DATA 1,1,1
290 DATA start of lines
300 DATA 1,2,1,3,1,5,2,4,2,6,3,4,3,7,4,8,5,6,5,7,6,
8,7,8
310 DATA end of lines
320 END DEF

```

*** command in program

=====

Doit deze error-message gezien ?

Hij treedt onder andere op als U in Uw programma's regels wil verwerken als deze:

```
10 LOAD "program"
```

De BASIC-editor geeft prompt bovengenoemde foutcode en in principe zijn commando's dus niet in een BASIC-programma te verwerken.

Omdat ik niet tot de mensen behoor die zich gemakkelijk ergens bij neerleggen, heb ik hier ook lak aan. Het volgende zal deze beperking namelijk opheffen.

Tik in program 0:

```
10 PRINT "program"
```

vervolgens:

```
EDIT 1
```

vervolgens tikt U program 1 in:

```
10 FOR A=4800 TO 4820
20 PRINT A,PEEK(A)
30 NEXT
```

vervolgens:

```
RUN
```

en dit is het resultaat:

4800	5
4801	87
4802	79
4803	82
4804	68
4805	36
4806	206
4807	205
4808	1
4809	16
4810	10
4811	0
4812	0
4813	96
4814	56
4815	128
4816	7
4817	112
4818	114
4819	111
4820	103

56 is de code voor het statement PRINT

Op de plaats waar het PRINT-statement staat, komt het LOAD--
commando door middel van:

```
POKE 4814,43
```

Ga terug naar PROGRAM 0 en zie daar:

```
10 LOAD "program"
```

U kunt nu verder gaan met de rest van Uw programma. Veel genot
wens ik U toe met deze truc.

Bas de Kruyff.